Simulated annealing for manufacturing systems layout design

Abdelghani Souilah

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Simulated annealing for manufacturing systems layout design

Abdelghani SOUILAH

N° 1909
Mai 1993

PROGRAMME 5

Traitement du signal,
automatique et
productique

Rapport
de recherche

1993

SIMULATED ANNEALING FOR MANUFACTURING SYSTEMS LAYOUT DESIGN
(LE RECUIT SIMULÉ POUR L'AGENCEMENT DES SYSTÈMES DE PRODUCTION)

Abdelghani SOUILAH
Projet SAGEP / INRIA
Technopôle Metz 2000, 4 rue Marconi, 57070 Metz, France
Tel.: (33) 87. 20. 35. 10 - Fax: (33) 87. 76. 39. 77
e-mail: souilah@ilm.loria.fr

SUMMARY: In this report, we first present the general Simulated Annealing (SA) algorithm. We then show how it has been used to group resources into manufacturing cells, to design the intra-cell layout, and to place the manufacturing cells on the available shop-floor surface. Some numerical examples are used to illustrate these approaches.
KEY WORDS: Simulated annealing, Manufacturing system layout, Optimization, Cellular manufacturing systems.

RÉSUMÉ: Dans ce rapport, nous présentons l'algorithme du recuit simulé. Nous montrons ensuite comment il a été utilisé (i) pour grouper les ressources en cellules de production, (ii) pour faire l'agencement des ressources à l'intérieur des cellules et enfin (iii) pour placer les cellules sur la surface libre de l'atelier. Quelques exemples illustratifs sont donnés pour chacune de ces applications.
MOTS CLÉS: Recuit similé, Agencement des systèmes de production, optimisation, systèmes cellulaires de production.

1. INTRODUCTION

A new four-stage approach to design an "optimal" layout is presented in this paper. The first stage called pre-processing stage transforms the problem at hand into a mono manufacturing process problem, i.e. a problem in which only one manufacturing process is used to manufacture a part, among alterantives. The main activities performed at the pre-processing stage consists in: (i) allocating the product to the resources (i.e. by sharing out the quantity of each part type to be manufactured among the available resources), and (ii) locating the buffers in the system. This problem is out of the scope of this paper.

The second stage consists of grouping physical facilities into cells in order to minimize the inter-cell traffic, taking into account some constraints[7]. This stage is called Manufacturing cell Design.

The aim of the third stage is to arrange the physical resources inside the cells. This task is performed in two steps. In the first step, an expert system is used to select the material handling system, from which we derive the cell type (several approaches are proposed in references 8, 11 and 6). The second step consists of solving the assignment problems in order to find the best location of each resource (This problem is detailed in the references 9 and 13). This stage is referred to as intra-cell location.

The fourth stage is devoted to the location of cells in the shop-floor in order to minimize the operating cost of the overall system. This operating cost is expressed as the sum of the products (distance between cells x material flow).

Figure 1 summarizes the above strategy. The refinement stage allows the user to modify the manufacturing layout in order to comply with some qualitative criteria.

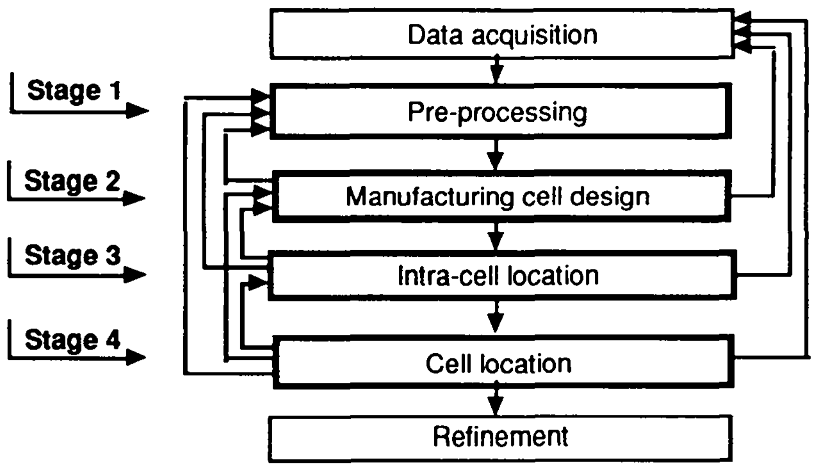


Fig. 1: The strategy for cellular manufacturing system design

The remainder of the paper is organized as follows. Section 2 introduces SA as an approach to solve the combinatorial optimization problems. Cell design, intra-cell layout and inter-cell layout problems are respectively developed in sections 3, 4 and 5. The final section presents some conclusive remarks.

2. THE SIMULATED ANNEALING ALGORITHM

SA is well adapted to combinatorial optimization problems[4]. It has been introduced by Kirkpatrick, Gelatt and Vecchi in 1983[10], as an application of the analogy between statistical mechanics and combinatorial optimization.

SA is a step-by-step method which could be considered as an improvement of the local optimization algorithm. The local optimization algorithm proceeds by generating, at each step of the computation, a solution in the neighbourhood of the previous one. If the value of the criterion corresponding to the new solution is better than the previous one, the new solution is selected, otherwise it is rejected. In both cases, we restart the process by choosing a solution in the neighbourhood of the solution at hand. The algorithm stops either when it is no more possible to improve the solution or the maximal number of trial decided by the user is reached. The drawback of the local optimization algorithm is that it terminates at a local minimum which depends on the initial solution and which may be far from a global minimum.

The SA algorithm allows not to be entrapped in a local optimum. The difference with the local optimization algorithm is that a solution B derived from a solution A is not only accepted if B is better than A according to the criterion, but it may also be accepted with a given probability if B is worse than A. This probability is equal to $\exp(-df/T)$, where **T** is a given parameter called temperature which decreases with the number of trials, and $df = f(B) - f(A) > 0$, where $f(\cdot)$ is the criterion. This is called the Metropolis acceptance rule. This acceptance rule implies that: (i) the smaller the increase of the criterion value, the more likely the new solution is selected, and (ii) the lower the value of T (i.e. the greater the number of trials), the less likely the new solution is selected.

The SA algorithm starts with an initial solution and a relatively high value of **T** to avoid being prematurely entrapped in a local optimum. At each step of the computation the algorithm generates several solutions in the neighbourhood of the previous one and decreases the temperature. A new solution is chosen at random among the ones which have been previously generated. This new solution is selected or not according to the Metropolis acceptance rule. The process restarts with the new solution (if accepted) or with the previous one (if the new solution has been rejected). Several tests can be applied to stop the process, for instance: number of trials, minimal value of **T**, minimal mean value of the improvement of the criterion during the last **n** trials. The temperature decreasing process is explained in section 2.1.

In the general SA algorithm illustrated hereafter we have two nested loops: the outer loop controls the temperature decreasing process and the inner loop controls the number of feasible solutions to generate at a given temperature, called the epoch length.

The processes used to generate the initial solution, a solution belonging to the neighbourhood of a given solution, and to compute the value of the criterion, depends on the problem at hand. We explain these processes for each of the problems developed hereafter.

The general simulated annealing algorithm

```
Select an initial solution I from the feasible set of solutions S;
Choose an initial temperature T0>0;
Choose the temperature decreasing process;
Choose the epoch length function;
Set temperature change counter  t=0;
Repeat
        Set the epoch length counter  n=0;
        Repeat
                Generate solution j in the  neighbourhood of I;
                Compute   df= f(j) - f(i);
                if  df<0  then  i:=j
                else if  random (0,1) < exp(-df/T)  then   i := j;
                n := n+1;
        until  n = N;  "Number of consecutive trials for
                        which the temperature is T"
        t := t + 1;
        Compute the new temperature T;
until  stopping criterion true.

Note:   The process to choose the control parameters is
        presented in the next section.
```

2.1. The simulated annealing control parameters

Four choices must be made when implementing an SA approach, namely: (i) the initial temperature T_0, (ii) epoch length function, (iii) the temperature decreasing process, and (iv) the stopping criterion.

2.1.1. Initial temperature

Kirkpatrick et al.[10] have proposed an initial temperature $\mathbf{T_0}$ large enough so that essentially all the solutions are accepted at the first stage of the annealing process with a probability $\mathbf{P_0} = 0.8$.

2.1.2. Temperature decreasing function

Remember that the temperature is used to compute the acceptance probability of a solution which is worse than the previous ones. Several functions have been proposed in the literature[5], for instance:

a. Arithmetic function $T_k = T_{k-1}$ - Cte.;
b. Geometric function $T_k = a_k \times T_{k-1}$ [a_k usually constant and less than 1];
c. Inverse function $T_k = Cte/(1 + k)$;
d. Logarithmic function $T_k = Cte/(Log(1 + k))$.

For solving the layout problems, we used a function which appears to be very common in the current literature. It is the function "b" where $\mathbf{a_k}$ is a constant less than (but close to) 1. Typical values for $\mathbf{a_k}$ are chosen between 0.85 and 0.95 (see reference 12).

2.1.3. Epoch length

Let $\mathbf{N_k}$ be the epoch length.(i.e. the number of trials to be performed with the same temperature value). Some functions are proposed in the literature[15] to define $\mathbf{N_k}$:

a. Constant N_k = Cte.;
b. Arithmetic $N_k = N_{k-1}$ + Cte.;
c. Geometric $N_k = N_{k-1} / a$, where **a** is a constant less than 1;
d. Logarithmic $N_k = Cte / Log(T_k)$;
e. Exponential $N_k = (N_{k-1})^{(1/a)}$, where **a** is a constant less than 1.

In our implementations the epoch length is controlled by two parameters: the maximal allowable numbers of times the new solution is accepted and rejected per epoch. We compute these values for a constant value of the temperature, and if one of these numbers, or both, become equal to their maximal allowable value, we say that the equilibrium state is established and we reduce the temperature in order to start a new epoch.

2.1.4. Stopping test

The last detail which should be defined is the stopping test, which will specify when the system is "frozen". Several tests are available, for instance:

a. a given total number of steps have been performed,

b. The required number of acceptances for a given number of trials has not been obtained.

c. a given final temperature is reached,

We use this last test in our applications.

3. CELL DESIGN

The goal of cell design is to group the machines into cells in order to minimize the inter-cell traffic, knowing that the number of machines in each cell is limited. The cell design is a combinatorial problem for which several heuristics have been proposed from McAuley (1972) to Harhalakis et al. (1990). We present in this section a heuristic algorithm based on SA. The problem is stated in section 3.1 and solved in section 3.2. A numerical example is provided in appendix.

3.1. Statement of the Problem

Let $\boldsymbol{M} = \{M_1, M_2, ..., M_q\}$ be the set of machines, $\boldsymbol{P} = \{P_1, P_2, ..., P_p\}$ the set of part types which can be manufactured using $\boldsymbol{M}$. For each part type P_i ($i = 1, 2, ..., p$) we define:

(i) the unique sequence of machines visited by the product in order to be manufactured. This sequence, called routing, is written as : $R_i = \langle M_{i,1}, M_{i,2}, \ldots, M_{i,s_i} \rangle$

where $M_{i,j} \in \boldsymbol{M}$ for $i = 1, 2, ..., p$ and $j = 1, 2, ..., s_i$;

s_i is the length of the routing of P_i.

(ii) the average rate r_i of part types P_i to be produced per unit of time.

For each $(P_i, M_k, M_l) \in \boldsymbol{P} \times \boldsymbol{M} \times \boldsymbol{M}$, we denote by:

$v_{i,k,l}$ the number of times M_k immediately follows M_l and M_l immediately follows M_k in the routing of P_i.

For each pair $(M_k, M_l) \in \boldsymbol{M} \times \boldsymbol{M}$, the traffic between M_k and M_l is defined as:

$$t_{kl} = \sum_{i=1}^{p} r_i \times v_{i,k,l} \tag{1}$$

Let $\boldsymbol{C} = \{C_1, C_2, ..., C_w\}$ be a partition of $\boldsymbol{M}$, then:

$$\bigcup_{i=1}^{w} C_i = \boldsymbol{M} \quad \text{and} \quad \bigcap_{i=1}^{w} C_i = \varnothing \tag{2}$$

Let $E_c(\boldsymbol{C}) = \{(m,n) \ / \ m \in C_k \text{ and } n \in C_l;\ k, l = 1, 2, \ldots, w \text{ and } k \neq l\}$ (3)

be the set of machine pairs (m, n) which are in different cells C_k and C_l of $\boldsymbol{C}$.

The total inter-cell traffic related to partition $\boldsymbol{C}$ is:

$$T_c(\boldsymbol{C}) = \sum_{(m,n)\in E_c(\boldsymbol{C})} t_{m,n} \tag{4}$$

If $\boldsymbol{S}$ is the set of partitions of $\boldsymbol{M}$ verifying the constraints which are:

(i) $\forall\, \boldsymbol{C} \in \boldsymbol{S}$ and $\forall\, C_i \in \boldsymbol{C}$ $(i = 1, 2, \ldots, w)$, $\text{card}(C_i) \leq N_i$: N is the upper bound on the size of the cells,

(ii) Some machine pairs (M_i, M_j) must obey to the following constraints:

$$\forall\, \boldsymbol{C} \in \boldsymbol{S},\ (i, j) \in E_c(\boldsymbol{C}) \tag{5}$$

$$\text{or } \forall\, \boldsymbol{C} \in \boldsymbol{S},\ (i, j) \notin E_c(\boldsymbol{C}) \tag{5'}$$

which means that machine i and j **must** be located in different cells (or in the same cell),

(iii) For other machine pairs (i, j) it is **preferable** to verify (5) or (5').

The problem to be solved consists of finding a partition $\boldsymbol{C}^* \in \boldsymbol{S}$ of $\boldsymbol{M}$ such that:

$$T_c(\boldsymbol{C}^*) = \operatorname*{Min}_{\boldsymbol{C}\in\boldsymbol{S}} T_c(\boldsymbol{C}) \tag{6}$$

In order to take into account constraint (iii), we penalize the objective function whenever it is violated. The new expression of the total inter-cell traffic is:

$$T_c(\boldsymbol{C}) = \sum_{(m,n)\in E_c(\boldsymbol{C})} t_{m,n} + k_{m,n} \times \alpha_{m,n} \times \max_{m,n} (t_{m,n}) \tag{7}$$

$$k_{m,n} = \begin{cases} 0 & \text{if the constraint between } M_m \text{ and } M_n \text{ is respected,} \\ 1 & \text{otherwise.} \end{cases} \tag{8}$$

$\alpha_{m,n}$ is called weight and it represents the importance given to the constraint between M_m and M_n. Its value is taken from Table 1.

Constraint designation	*Weight*
Very important	2
Important	1
Quite important	0.5
Weak	0.25
Very weak	0.125

Tab. 1: The weight scale of the constraint related to the proximity of machines

3.2. Solution of the problem

This problem is solved using the SA approach presented in section 2. We explain hereafter the points of the SA algorithm specific to our application.

3.2.1. The initial configuration

The initial feasible configuration is generated at random. The initial number of cells is equal to:

$$\begin{cases} \mathrm{card}(\boldsymbol{M}) \ / \ N & \text{if } \mathrm{card}(\boldsymbol{M}) \text{ is a multiple of } N \\ \lceil \mathrm{card}(\boldsymbol{M}) \ / \ N \rceil + 1 & \text{otherwise} \end{cases} \tag{9}$$

where $\lceil x \rceil$ is the smallest integer greater than or equal to x ($x \in \Re^{+}$).

While designing the initial partition, each machine M_i, i = 1, 2, ..., m, is assigned to a cell chosen at random, with respect to constraints (i) and (ii). At the end of this operation, the criterion value corresponding to the partition is computed using (7).

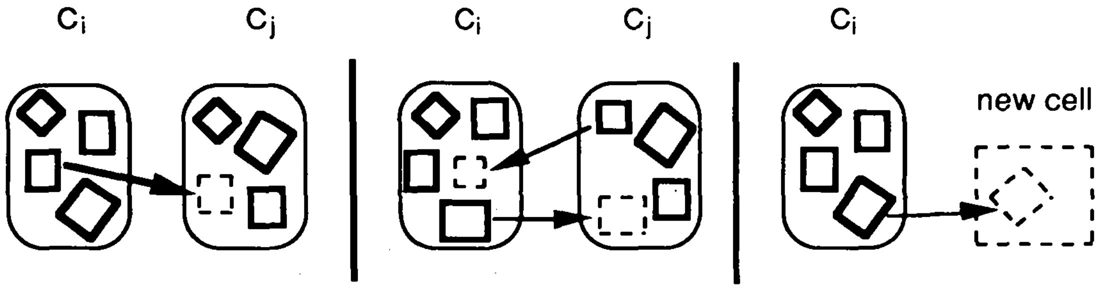
3.2.2. The elementary transformations

It has been shown[3] that the efficiency of SA depends on the neighbourhood structure or the elementary transformation process used. In general, an elementary transformation process which imposes a smooth graph of the prospected feasible solutions' cost is preferred to a bumpy graph where there are many deep local minima.

The transition from a partition $\boldsymbol{C}$ to another partition $\boldsymbol{C'}$ belonging to the neighbourhood of $\boldsymbol{C}$, is realized using one of the following elementary transformations, chosen at random (see Figure 2):

a. Removing a machine from a cell and placing it in another cell.
b. Swapping two machines belonging to different cells.
c. Moving a machine from its cell to a newly generated cell.

The cells are also chosen at random and the transformation are not realized if they do not respect the constraints (i) and (ii). If during the process a cell becomes empty, it is automatically ignored.



a. Removing a machine b. Swapping two machines c. Moving a machine to a new cell

Fig. 2: The elementary transformations

4. INTRA-CELL LAYOUT DESIGN

As explained in the introduction, this problem is solved in three steps. We start by giving a brief presentation of the results provided by the two first steps. They are used in the third step, which is the principal part of the current section. The second step of the intra-cell layout design is an assignment problem, followed by the definition of the cell dimensions and the location of its entrance and exit. A numerical example is given in appendix.

4.1. General framework

Using an Expert System, the system starts by selecting the Material Handling System (MHS) to be used in the cell. It is selected according to the parts to be manufactured, the machines and the cell characteristics. MHSs commonly used inside a cell are: robots, gantry robots, conveyors, or AGVs (see Figure 4). Then the system select a layout type according to the selected MHS and the parts and machines characteristics. This section is also made using an Expert System. The usual layout types are: linear single-rows, circular single-rows, double-rows, and the multi-rows (see Figure 5). A free layout type can be introduced by giving the possible locations of the machines (for more informations about the ESs see references 8 and 6). As soon as the MHS and the layout type of the cell are selected, the assignment of machines to sites starts. This problem is presented in the next sub-section. Figure 3 illustrates the general framework of the intra-cell layout design step.

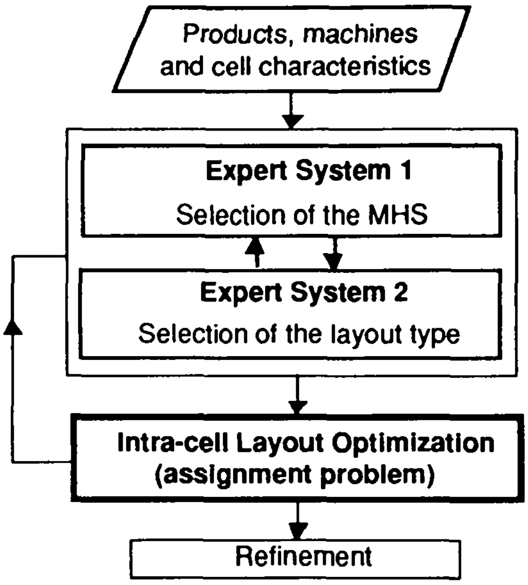


Fig. 3: The general framework of the Intra-cell layout design

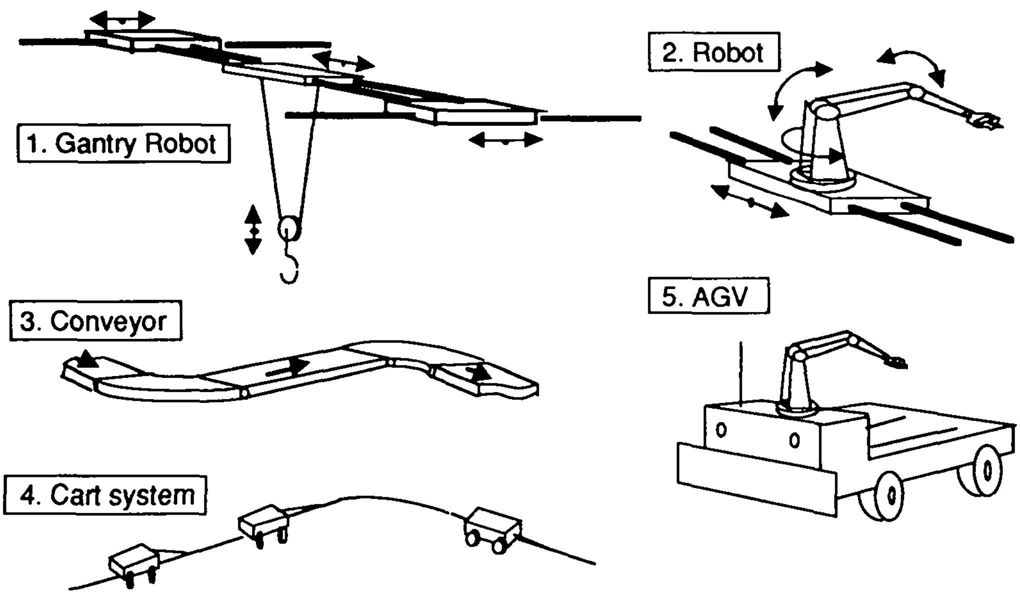


Fig. 4: MHSs used in a cell

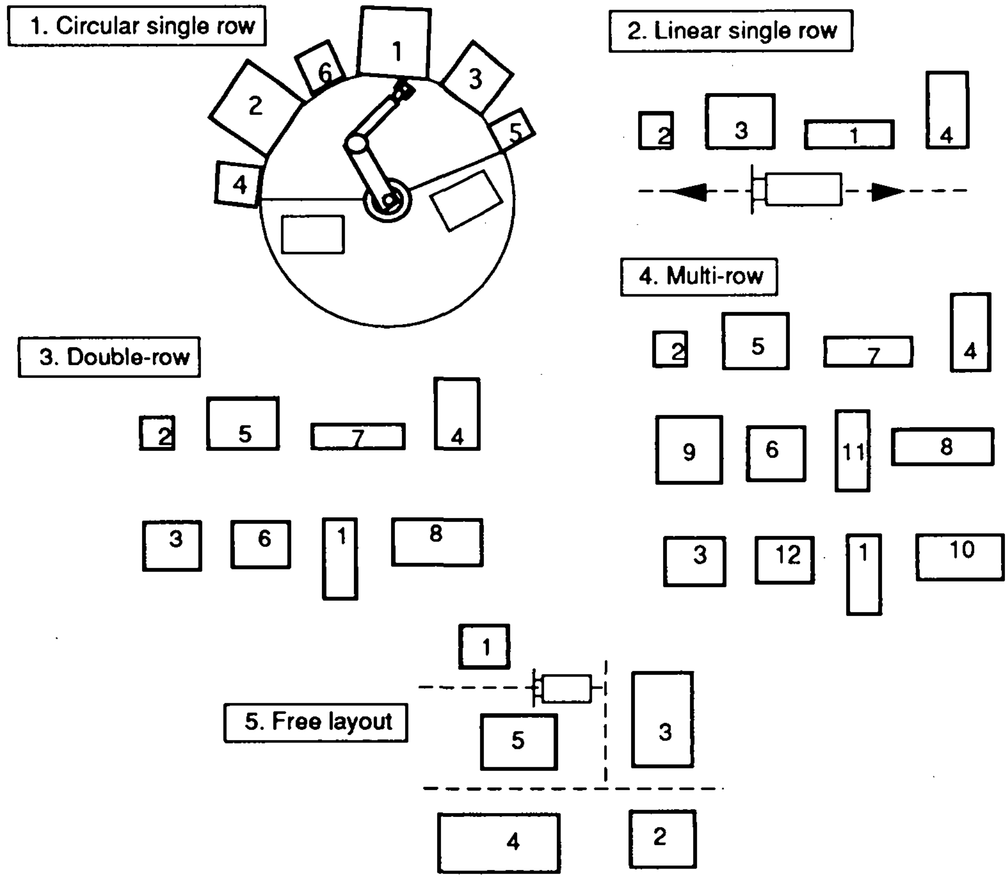


Fig. 5: Cell layout types

4.2. Statement of the optimization problem

Let C be a manufacturing cell which includes m machines, say $C = \{M_1, M_2, ..., M_m\}$ ($C \subset \boldsymbol{M}$). C is dedicated to the manufacturing of a part family Pf composed of a set of n part types, $Pf = \{P_1, P_2, ..., P_n\}$ ($Pf \subset \boldsymbol{P}$). We use the notations of section 3, and further introduce the following definitions:

For each triplet $(P_k, M_i, M_j) \in Pf \times C \times C$, we denote by $u_{k,i,j}$ the pallet size of the part type P_k transported from machine M_i to machine M_j.

The material flow from M_i to M_j is defined as follows:

$$f_{i,j} = \sum_{k=1}^{n} r_k \times \frac{v_{k,i,j}}{u_{k,i,j}} \tag{10}$$

For each pair of machines $(M_i, M_j) \in C \times C$, we define:

(i) the transportation cost of the part types per unit of distance, denoted by $c_{i,j}$ (all the parts use the same material handling system; thus the transportation cost per unit distance is the same for all of them);

(ii) the distance between the output of M_i and the input of M_j, denoted by $d_{i,j}$. This distance depends on both the layout type of the cell and the material handling system used inside the cell.

The site of a machine M_i is noted $s(M_i)$. The problem to solve is to find the best assignment of machines M_i ($i = 1, 2, ..., m$) to different sites, i.e. the assignment which minimizes the criterion:

$$\sum_{i=1}^{m}\sum_{\substack{j=1\\j\neq i}}^{m} c_{i,j} \times f_{i,j} \times d_{i,j} \tag{11}$$

This objective function is subject to:

(i) $\forall\ i, j \in \{1, 2, ..., m\}$ and if $i \neq j$, then $s(M_i) \neq s(M_j)$,

(ii) $\forall\ i, j \in \{1, 2, ..., m\}$ and $i \neq j$, then if M_i is adjacent to M_j, $\Delta_{i,j} \geq \Delta_{min}$, where:

$\Delta_{i,j}$ is the actual distance between the adjacent sides of M_i and M_j,

Δ_{min} is the minimal distance between the adjacent sides of M_i and M_j.

(iii) Some machines must be **definitely** assigned to a given site.

(iv) Some pairs of machines (M_i, M_j) need to be located far from (resp. close to) each other, with keeping between their centers a distance greater than or equal to $\Delta min_{i,j}$ (resp. less than or equal to $\Delta max_{i,j}$).

Constraint (iv) can be strong (i.e. if it does not hold the layout is rejected), or weak (i.e. if it does not hold the criterion value is penalized). We introduce then a parameter $\lambda_{i,j}$ which increases with the importance of the constraint between M_i and M_j.

In order to take into account the constraints (iv) when they are weak, we modify the objective function as follows:

$$\sum_{i=1}^{m}\sum_{\substack{j=1\\j\neq i}}^{m} c_{i,j} \times \left(f_{i,j} \times d_{i,j} + \phi_{i,j} \times \delta_{i,j}\right) \tag{12}$$

where :

$$\delta_{i,j} = \begin{cases} \Delta min_{i,j} - d_{i,j} & \text{if } M_i \text{ and } M_j \text{ have to be far from each other and } d_{i,j} < \Delta min_{i,j}, \\ d_{i,j} - \Delta max_{i,j} & \text{if } M_i \text{ and } M_j \text{ have to be close to each other and } d_{i,j} > \Delta max_{i,j}, \\ 0 & \text{otherwise.} \end{cases} \tag{13}$$

$$\phi_{i,j} = \begin{cases} \lambda_{i,j} \times \max_{i,j}(f_{i,j}) & \text{if } \delta_{i,j} \neq 0, \\ 0 & \text{otherwise.} \end{cases} \tag{14}$$

$\phi_{i,j}$ is the weight associated to the constraint between M_i and M_j,

$\delta_{i,j}$ measures the importance of the constraint violation.

The term ($\phi_{i,j} \times \delta_{i,j}$) is added in order to penalize the criterion value if the weak proximity constraint between M_i and M_j is not respected. The values of the parameter $\lambda_{i,j}$ are chosen from Table 1.

4.2. Solution of the problem

As for the previous one, this problem is solved using an SA algorithm. The parts of the algorithm which are specific to this application are discussed hereafter.

In order to make the discussion more understandable, we consider an example where the cell type is a multi-row, and the material handling system is a gantry robot. This case generalizes the single and the double-row layout types. We simplify the computation by assuming that the input and output points of each machine are both on its geometrical center. This assumption does not introduce any restriction.

The variables to be computed are: the distances between the input and output of the machines denoted by d_{ij}. They are used to compute of the criterion value (12), and the clearance distance between two adjacent machines (see constraint (ii)).

The positions of the machines in the cell are given by the coordinates of their centers with respect to an orthogonal system (see Figure 6). The distance between machines M_i (x_i, y_i) and M_j (x_j, y_j) is the Euclidean distance:

$$d_{i,j} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{15}$$

We denote by l_i (resp. w_i) the dimension of the side of machine M_i parallel to the x-axis (resp. y-axis). The clearance between the adjacent sides of two machines M_i and M_j are:

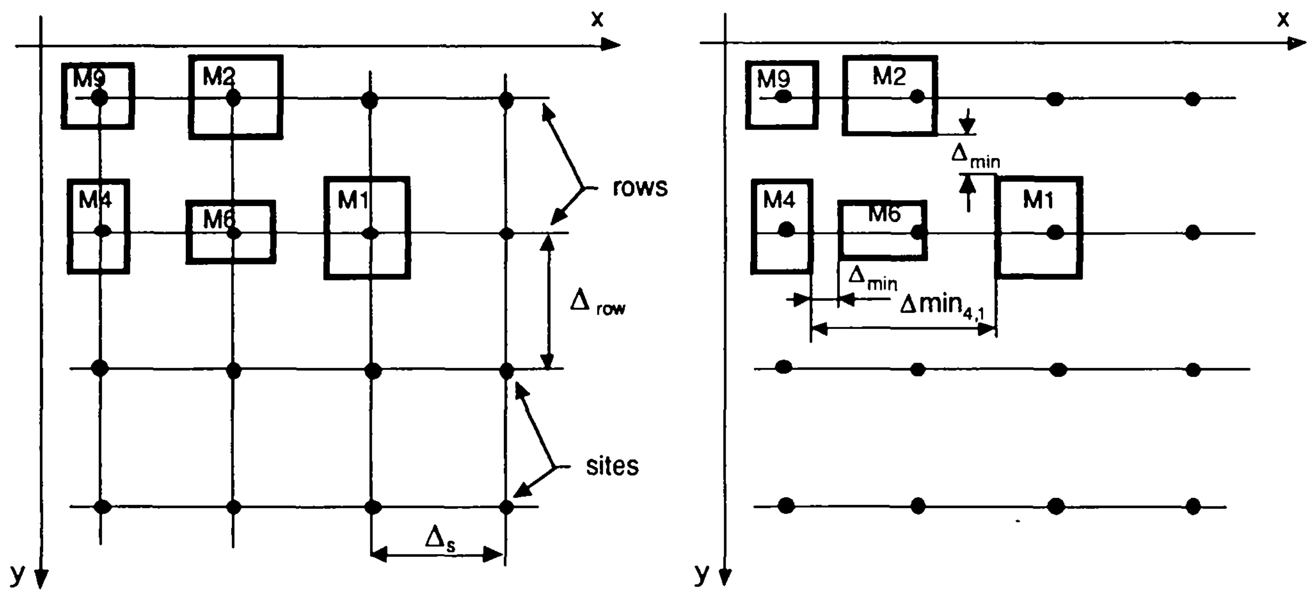
$$\begin{cases} \Delta x_{i,j} = |x_i - x_j| - \frac{1}{2}(l_i + l_j) & \text{if the sides are parallel to the y-axis,} \\ \Delta y_{i,j} = |y_i - y_j| - \frac{1}{2}(w_i + w_j) & \text{if they are parallel to the x-axis.} \end{cases} \tag{16}$$

4.2.1. Initial configuration

The design of the initial configuration is made in two steps (see Figure 6). The rows of the considered cell are initially separated by the same distance Δ_{row}, and the sites located on the same row also separated by the distance Δ_s:

$$\Delta_s = \Delta_{row} = \max\left\{\Delta_{min}, \max_{i,j}\left\{\Delta min_{i,j}\right\}\right\} + \max_i\{\max(l_i, w_i)\} \ (i, j = 1, 2, ..., m) \tag{17}$$

In this case, these two distances are the same because we use a gantry robot as the MHS. So the machines are separated by the same distance in the two directions of the plane. In the first step, the machines are assigned to the sites at random taking into account the constraints (i) and (iii). In the second step, they are brought together according to constraints (ii) and (iv). These operations are followed by the computation of the distances between the machines and the cost of this initial configuration using (12).



Step 1. Assignment of the machines to the sites
(The main grid)

Step 2. Bringing the machines together

Fig. 6: Design of an initial configuration

4.2.2. Elementary transformations

There are two steps in the neighbourhood process. In the first step, the sequences of machines on the rows of the main grid (see Figure 7) are modified by means of one of the following operations:

1. Swappiing two machines chosen at random (Figure 7.1);

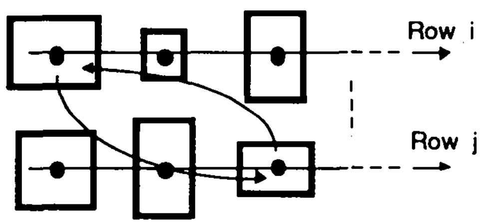
2. moving one machine chosen at random to a free site at the end of the corresponding row (Figure 7.2);

3. Moving one machine chosen at random to an occupied site (Figure 7.3),

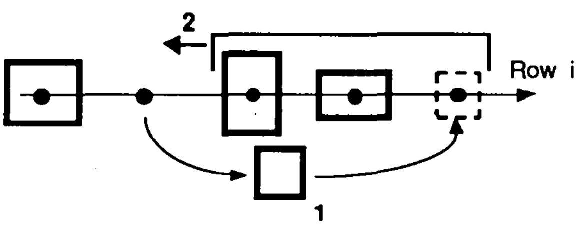
    • by applying a circular permutation, if the old and the new sites are on the same row.

or • by applying translations, if the machines are on different rows.

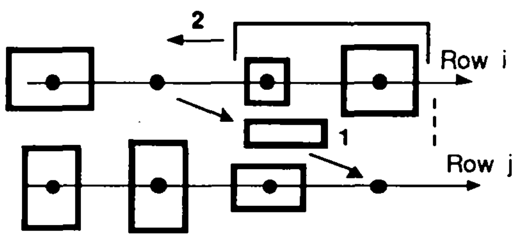
In the second step, machines are brought together in order to fit constraints (ii) and (iv). The distances between machines are obtained using relation (15). They are used to compute the criterion value using relation (12).



7.1. Permuting two machines

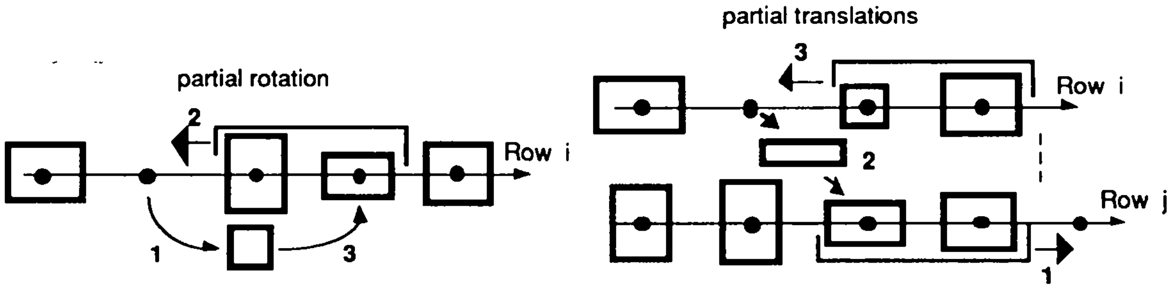


a. On the same line



b. To an other line

7.2. Moving a machine to a free site



a. On the same line

b. To an other line

7.3. Moving a machine to an occupied site

Fig. 7: Elementary transformations

4.2.3. Dimensions of the cell

Cells are assumed to be rectangular shaped. The rectangle which represents a cell is the smallest rectangle containing the cell surrounded by a security area (see Figure 10).

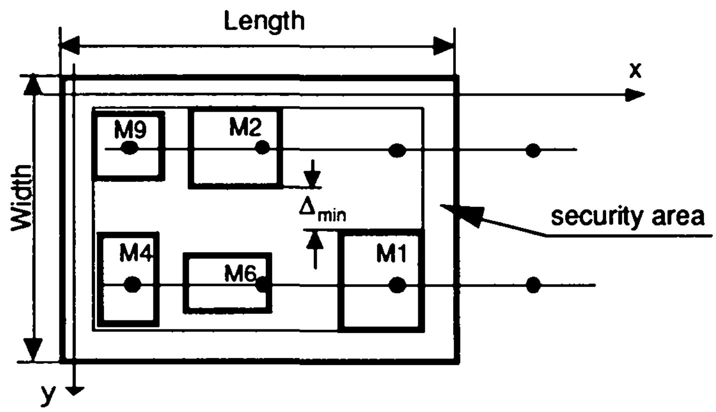


Fig. 8: A cell

4.3.4. Location of the cell input and output

We propose two possible approaches for locating the input and the output of a cell:

4.3.4.1. First approach

The first approach applies when the input and the output of the cell are known. In this case, the input and the output of the cell are considered as machines and added to the manufacturing processes of the products, respectively at their beginning and at their end. Then, the new material flow matrix is computed, and the problem is solved using formulation (11) taking into account constraints (i) and (ii).

4.3.4. 2. Second approach

The second approach is used when the input and the output have to be respectively assigned to one of the pre-defined sites located at the limits of the cell. The first step consists in solving the intra-cell layout problem without taking into account the output and the input of the cells. The selection of the input and the output sites is then made by minimizing:

$$\sum_{i=1}^{m} c_{E,i} \times f_{E,i} \times d_{E,i} + c_{i,S} \times f_{i,S} \times d_{i,S} \qquad (20)$$

where E is the input of the cell and S its output.

5. CELL LOCATION

The shop-floor is defined by its geometry, i.e. its surface limits, the physical obstacles (halls, power sources, forbidden areas, ...), and the positions of the inputs and the outputs. The buffers between the cells, the parking places of the AGVs and the tool magazines are considered as cells. A single machine can be considered as a cell too. We summarize the previous information in Figure 9. A numerical example is proposed in appendix.

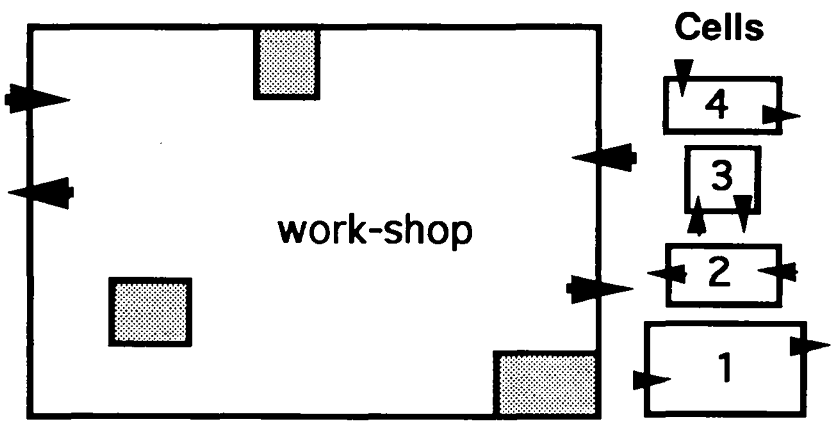


Fig. 9: The manufacturing system and the cells to be laid out

The products manufactured in the system are characterized by their manufacturing processes or routings. The matrix of the material flow between cells is computed by considering the part type routings, the composition of the cells, the average production ratios of the part types and their lot sizes. Thus, this matrix represents the dynamics of the system[14].

5.1. Statement of the problem

Mathematically, the inter-cell layout problem consists in locating some rectangular shaped cells on the free surface of the shop-floor, in order to minimize the sum of the products (Flow x Distance) between the pairs of cell outputs and inputs, taking into account the constraints. Note that the input (resp. output) of the shop-floor is considered as the output (resp. the input) of a cell.

5.1.1. Objective function

A possible formulation of the problem is as follows:

$$\text{Minimize } \sum_{i=1}^{Nd} \sum_{j=1}^{Na} T(i,\ j) \times D(i,\ j). \tag{21}$$

where:

Nd is the number of cell outputs,

Na is the number of cell inputs,

T(i, j) is the material flow cost per unit of distance between the cell output i and the cell input j,

$$T(i,\ j) = \sum_{k=1}^{p} \left(c_k \times r_k \times \frac{V_{k,i,j}}{L_{k,i,j}} \right) \tag{22}$$

where:

c_k is the transportation cost of part type P_k per unit of distance,

r_k is the ratio of parts P_k to be produced,

$V_{k,i,j}$ is the number of times the cell input j follows cell output i in the routing of P_k,

$L_{k,i,j}$ is the lot size of P_k between cell output i and cell input j.

D(i, j) is the distance covered by the transportation system from cell output i to cell input j.

5.1.2. The constraints

The objective function is subject to two types of constraints.

a. The strong constraints which cannot be violated.

b. The weak constraints which, if violated, lead to a penalty.

5.1.2.1. Strong constraints

Strong constraints are non-overlapping constraints, constraints to make sure that machines are far enough from each other and strong assignment of some cells to given locations in the shop-floor.

5.1.2.2. Weak constraints

These constraints give the desired minimal or maximal distance between cells. If such a constraint is violated, the objective function is penalized as shown hereafter:

$$\text{Minimize} \quad \sum_{i=1}^{Nd} \sum_{\substack{j=1 \\ j \neq i}}^{Na} [T(i, j) \times D(i, j) + TC(i,j) \times DC(P_i, P_j)] \tag{23}$$

where, for i, j = 1, 2, ..., Nc, $i \neq j$:

$$DC(P_i, P_j) = \begin{cases} Dmin_{i,j} - D(P_i, P_j) & \text{if cells i and j need to be far from each other and } D(P_i, P_j) < Dmin_{i,j}, \\ D(P_i, P_j) - Dmax_{i,j} & \text{if cells i and j need to be close to each other and } D(P_i, P_j) > Dmax_{i,j}, \\ 0 & \text{otherwise.} \end{cases} \tag{24}$$

$$TC(i,j) = \begin{cases} \lambda_{i,j} \times \max_{i,j}(T(i, j)) & \text{if } DC(P_i, P_j) \neq 0, \\ 0 & \text{otherwise.} \end{cases} \tag{25}$$

$Dmin_{i,j}$ (resp. $Dmax_{i,j}$) represents the minimal (resp. maximal) desired distance between cells C_i and C_j. (P_i,P_j) is the pair of points belonging to (C_i, C_j) such that:

$$D(P_i, P_j) = \min_{\substack{A \in C_i \\ B \in C_j}} D(A, B) \tag{26}$$

$\lambda_{i,j}$ is the weight of the constraint. Its value, chosen in Table 1, represents the importance given to the constraint.

5.2. Solution of the problem

The solution of the problem is obtained by using an SA algorithm. Nevertheless, it is necessary to explain the points which are specific to this application.

5.2.1. Construction of an initial configuration

The first step of the algorithm consists in finding a feasible solution to the problem, i.e. a solution which verifies the strong constraints.This problem is similar to the two-dimensional packing or cutting problems which are NP-hard (for more details see references 1 and 2). The difficulty of this problem increases as the difference between the available surface of the shop-floor and the sum of the cells' surfaces decreases.

The initial configuration is either provided by the user, or provided by the user aided by the computer, or generated by the computer.

We propose a very simple heuristic, based on common sense. This heuristic is used when the initial configuration is totally or partially built by the computer (see Table 2).

5.2.2. Elementary transformations

At each iteration of the SA process, the choice of configuration belonging to the neighbourhood of the current one is made by applying one of the following transformations chosen at random:

a. Swapping two cells chosen at random. This is generally followed by a rotation and/or a translation of the cells in order to make the new configuration feasible (Figure 10.a).

b. Moving a cell chosen at random toward a free site of the shop-floor. In this case the system defines a free zone able to receive the cell, installs and adjusts this cell using rotations and/or translations (Figure 10.b).

c. Translating a cell, chosen at random, in its own neighbourhood (Figure 10.c).

d. Rotating a cell, chosen at random, in its own neighbourhood (Figure 10.d).

Order the cells according to the increasing value of the surfaces,

For each cell considered in the previous order:

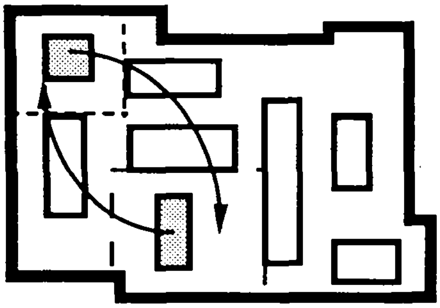
- find all the free zones able to contain the cell,
- select the free zone whose surface is the closest to the surface of the cell,
- install the cell.

End

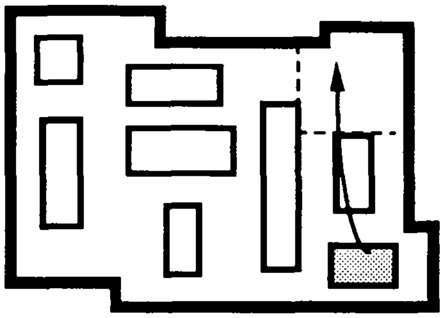
Tab. 2: Initial configuration building algorithm

5.2.3. Configuration criterion value

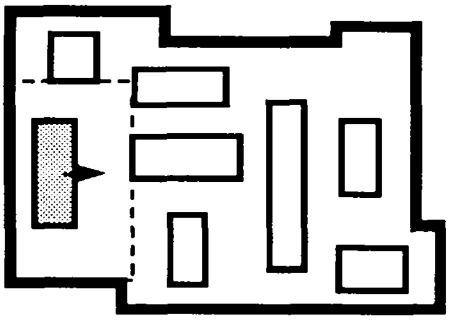
The criterion value of a given configuration is presented in (21). The computation of this criterion requires the lengths of paths joining all the cell outputs to the cell inputs The computation of these paths depends on the type of transportation system used. For instance, if the transportation system is a gantry robot able to perform two perpendicular motions at the same time, the paths are the straight lines joining cell outputs to cell inputs, and the the distance is the Euclidian distance. If the motions can not be performed simultaneously, the distance used is the Manhattan distance. The computation of these two distances is very easy compared to the case where the transportation system is an AGV: in this case, we have to compute the shortest path which avoid all the obstacles (see reference 14).



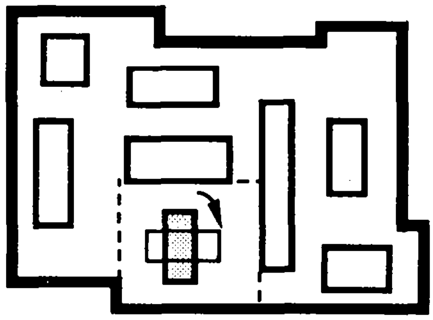
10.a. Swapping two cells



10.b. Moving a cell toward a free site



10.c. Translating a cell in its neighbourhood



10.d. Rotating a cell in its neighbourhood

Fig. 10: The elementary transformations

6. CONCLUDING COMMENTS

SA algorithm has, one more time, proved its efficiency to provide a near-optimal solution to combinatorial optimization problems. An SA approach presents several benefits: (i) it can by applied whatever the criterion and the constraints; (ii) it gives the possibility to solve large size problems; and (iii) it provides various good solutions starting from the same initial feasible solution. The major drawback is that there is no systematic procedure to obtain the best values of the control parameters for the problem at hand.

In this work we proposed a solution for the manufacturing systems layout design problem considered as a static problem . The dynamic aspect of the layout problem remains open. In particular, an ongoing research studies the stability of the layout when the environment changes (i.e. change of demand and/or of some of the resources).

REFERENCES

1. B.S. Baker, E.G. Coffman and R.L. Rivest, `Orthogonal packing in two dimensions', Society of industrial and applied mathematics journal, **9 (4)**, 846-855 (1980).

2. J.O. Berkey and P.Y. Wang, `Two-dimensional finite bin-packing algorithms', Journal of operational research society, **38 (5)**, 423-429 (1987).

3. V. Cerny, `Thermodynamical approach to the travelling salesman problem: An efficient simulation algorithm', Journal of optimization theory and applications, **45**, 41-51 (1985).

4. R.W. Eglese, `Simulated annealing: a tool for operational research', European journal of operational research **46**, 271-281 (1990).

5. S.B. Hamma et C. Bonnemoy, `La méthode du recuit simulé: Optimization globale dans R^{n}', APPII, **25**, 477-496 (1991).

6. T. Hamann, `Le problème d'agencement des ressources à l'intérieur des cellules des systèmes de fabrication', Ph. D Thesis, Metz University (France), 1992.

7. G. Haralakis, J. M, Proth and X. L. Xie, `Manufacturing cell design using simulated annealing: an industrial application' Journal of Intelligent Manufacturing **1**, 185-191 (1990).

8. S. Heragu and A. Kusiak, `Analysis of expert systems in manufacturing design', IEEE transactions on systems, man and cybernetics, **SMC-17 (6)**, 898-912 (1987).

9. F.S. Hillier & M.M. Connors, `Quadratic assignment problem algorithm and the location of indivisible facilities', Management Science, **13**, 42-57 (1966).

10. S. Kirkpatrick, C.D. Gelatt and M.P. Vecchi, `Optimization by simulated annealing', SCIENCE, **220 (4598)**, 671-680 (1983).

11. A. Kusiak, `Artificial intelligence and operations research in flexible manufacturing systems', INFOR, **25 (1)**, 2-11 (1987).

12. P. J. M. Van Laarhoven and E. H. Aarts, `Simulated annealing: Theory and Applications'. (Dordrecht: D. Reidel Publishing Co., 1987.

13. C.E. Nugent, T.E. Vollmann and R. Ruml. `An experimental comparison of techniques for the assignment of facilities to locations', Operations research, **16**, 150-173 (1968).

14. J.M. Proth and A. Souilah, `Near-Optimal Layout Algorithm Based on Simulated Annealing', International journal of Systems Automation: Research and Applications (SARA), **2**, 227-243 (1992).

15. J. Rose, W. Klebsch and J. Wolf, `Temperature Measurement and equilibrium dynamics of Simulated Annealing placement', IEEE Transactions on computer aided design, **9 (3)**, 253-259 (1990).

APPENDIX

(Numerical examples)

A. Cell design

The example proposed for the cell design step concerns a manufacturing system made of 23 machines which manufactures 22 types of products. We consider that the production ratio is the same for all the product types.

Table A1 gives the sequence of machines visited by the products. Each product visits at most 3 machines. The numbers in Table A1 represent the machines in the order they are visited.

	PRODUCT																					
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
1st mach.	18	20	12	12	19	22	20	7	7	3	14	21	4	9	21	10	14	5	22	18	13	1
2nd mach.	17	4	16	13	16	2	23	1	18	21	21	14	15	3	9	5	21	11	11	1	19	18
3rd mach.			19						7			21						10			13	

Tab. A1: The product routings

The cell design algorithm provided 7 cells. Table A2 gives the list of machines in each cell.

	CELLS						
	1	2	3	4	5	6	7
Machines contained in the cells	11 2 10 5 22	1 17 18 7	15 4 23 20	8	14 9 21 3	6	12 13 16 19

Tab. A2: The final solution

B. Intra-cell layout design

The intra-cell layout algorithm was applied to a cell composed of 8 machines which manufacture 17 product types. Table B1 provides the dimensions of the machines, the lot sizes for each product type, and the routing of each product type. The quantity is the quantity to be produced during a given period T. For instance, 10 products of type 2 have to be manufactured during period T. The pallet size is 1 (i.e. they are transfered in a lot of one in the cell) and products have to visit successively machines 4, 8, 7 and 8 again. The machines of this cell are unconstrained.

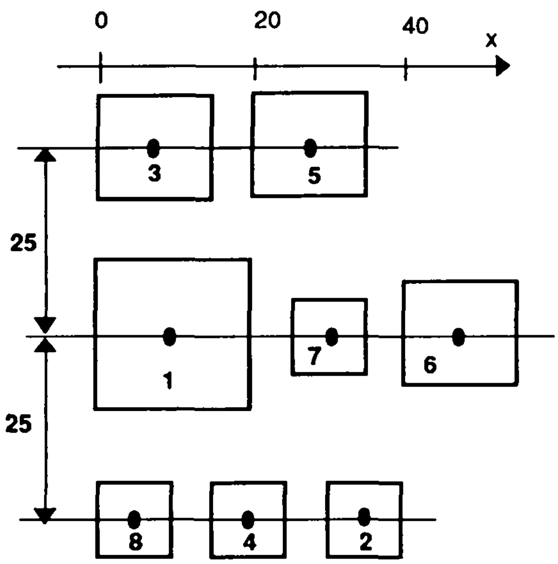
			PRODUCTS																	
			Product	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
MACHINES			Quantity	8	10	20	40	10	4	30	100	10	20	12	10	2	30	5	50	5
Mac.	Len. [dm]	Wid. [dm]	lot size	1	1	4	20	5	2	30	10	5	5	4	10	1	6	5	10	5
1	20	20	Routing	1			2,4			1,4		1		4	1		2			2
2	10	10					1	1,5	4			2,4		2		1				
3	15	15		2										1,3			1			
4	10	10			1	2		3	2	2	2		1							1
5	15	15				1				3		3	2,4		2			3	2	
6	15	15						2	3				3		3,5			2	1,3	
7	10	10			3			4	1						4	2				
8	10	10			2,4		3				1				6			1		

Tab. B1: Machines and products characteristics

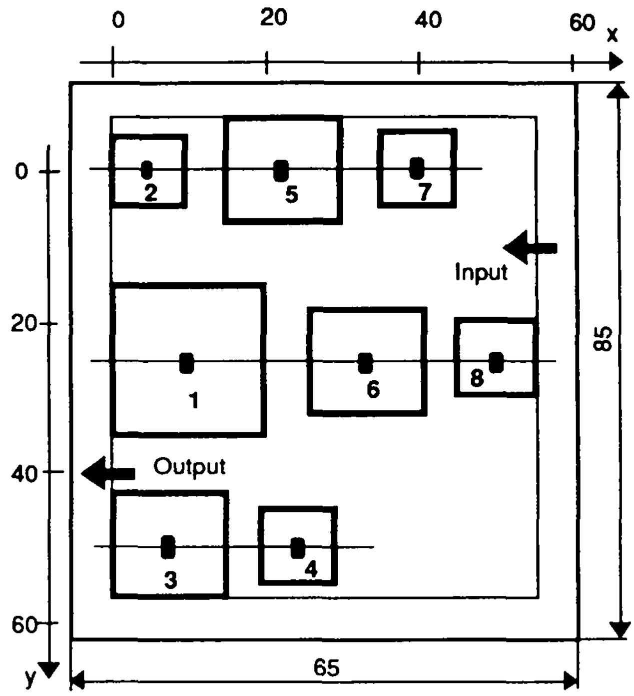
The first ES[6] led to a gentry robot as transportation system. The second ES[6] led to a multi-row structured cell. The maximal number of rows is fixed to 3, the maximal number of sites per row is 3 and the distance between two rows is fixed to 25 according to the machines' dimensions end to the minimal clearance between machines.

The criterion value corresponding to the initial layout (see Figure B1.a) is 3812.40. After 1145 trials realized during 20.36 sec. CPU (on a SUN sparc station 330), the system gives a final layout (see Figure

B1.b where are given the cell's dimensions and input:output positions) for which the criterion value is 2210.12 (an improvement of 42.03 %).



B1.a Initial intra-cell layout



B1.b Final intra-cell layout

Fig. B1: Intra-cell layout design: initial and final layouts

C. Cell location

The manufacturing system at hand is composed of 12 machines which manufacture 10 product types. The machines are grouped in the following 5 cells partition:

Cells	List of machines
1	1 12 4
2	2 9 6
3	7 10
4	3 11
5	5 8

Tab.C1: Cells composition

The shop-floor, represented in Figure C1, has two inputs and two outputs, which are located as shown in Table C3. The forbidden zones are located as shown in Table C2. We have to lay out the 5 cells described in Figure C2 and Table C4. Note that the positions are defined using an xy-system defined in Figure C1 and that the dimensions are multiples of a given unit distance [ul].

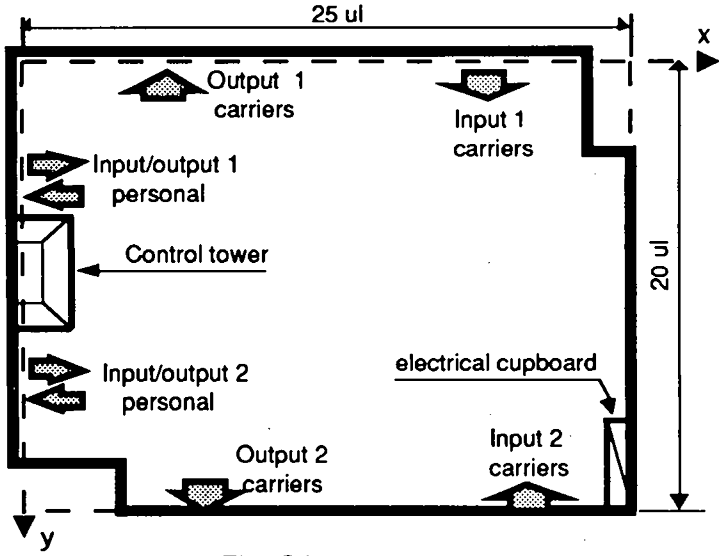


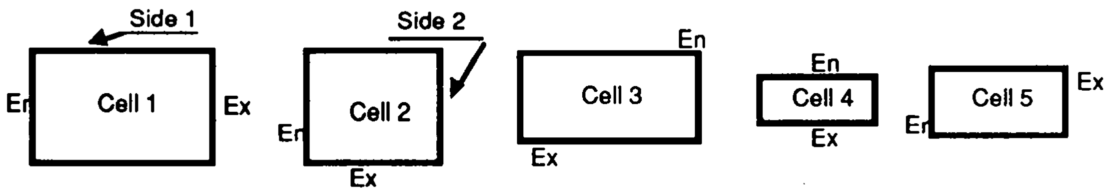
Fig. C1: The work-shop

Zone index	Side 1 dim.	Side 2 dim	position of the 1st vertex X	Y
1	2	4	24	1
2	2	5	1	8
3	2	4	24	17
4	5	2	1	19

Tab.C2: Dimensions of the forbidden zones

	X	Y
input 1	18	1
input 2	18	20
output 1	5	1
output 2	5	20

Tab.C3: Entrances and exits positions



Ex represents the output and **En** the input

Fig. C2: The cells

Cell index	Side 1 dim.	Side 2 dim.	Input pos. X	Input pos. Y	output pos. X	output pos. Y
1	8	5	0	3	8	3
2	6	5	0	2	3	5
3	8	4	8	0	1	4
4	5	2	3	0	3	2
5	6	3	0	3	6	1

Tab. C4: The cell dimensions

Constraints: Cell 5 contains machine 5 which is a cleaning and loading machine. This cell has to be installed near input 1 of the shop-floor, at a given position. We consider that the production ratios are identical, that the lot sizes and the transportation costs are equal to one. Table C3 gives the routings of the products. A shop-floor input or output equal to 0 means that the product does not come from or go to the outside of the shop-floor: it is the case for products assembled in the system or used as components for other products.

Product Idex	Routing	Entr. num.	Exit num.
1	5 3 11 1 12	1	1
2	2 6 7	0	0
3	2 11 9 6 10 7	0	2
4	6 2 12 11	2	2
5	5 8	1	0
6	1 12 4 7 10	2	1
7	4 1 6 11 3	2	2
8	5 8 2	1	0
9	9 2 11	0	1
10	6 11 12	2	0

Tab. C3: Product routings

The computed material flow matrix is given in Table C4:

	1	2	3	4	5	Ex1	Ex2
1	0	1	1	1	0	1	0
2	1	0	2	4	0	0	0
3	0	0	0	0	0	1	1
4	2	1	0	0	0	1	2
5	0	1	0	1	0	0	0
En1	0	0	0	0	3	0	0
En2	2	2	0	0	0	0	0

Tab. C4: The inter-cell material flow matrix

The initial layout proposed by the system is presented in Figure C3. The corresponding cost is 281.

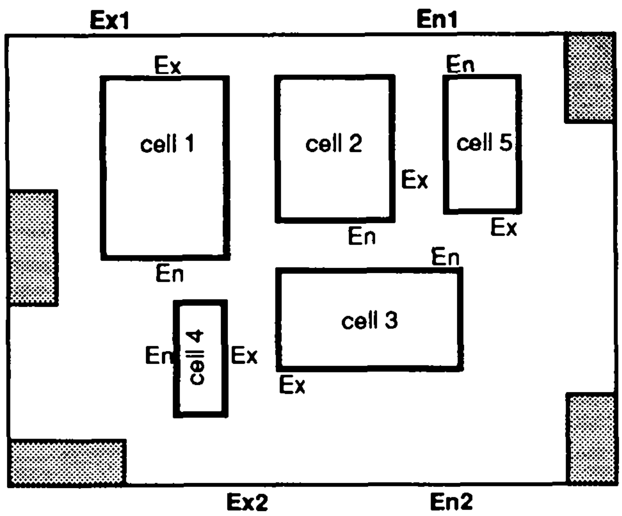


Fig. C3: The initial cell location

After 1058 trials (125 sec. CPU on a SUN sparc station 330), SA gives the following layout for which the criterion value is 222. The improvement according to the initial configuration is 21%.

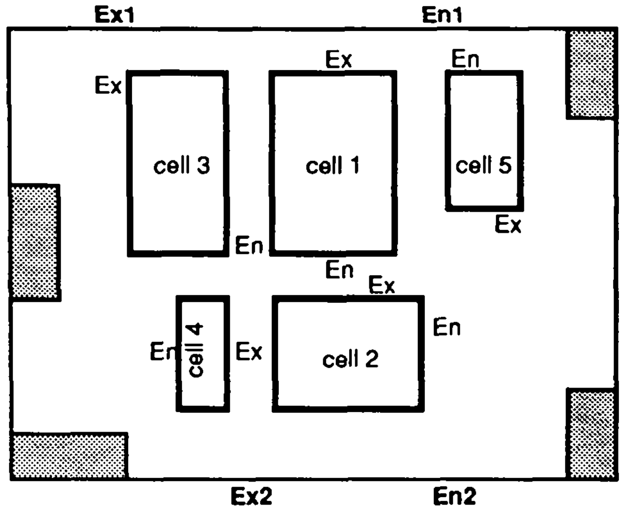


Fig. C4: The final cell location

Unité de Recherche INRIA Lorraine
Technopôle de Nancy-Brabois - Campus Scientifique
615, rue du Jardin Botanique - B.P. 101 - 54602 VILLERS LES NANCY Cedex (France)
Antenne de Metz
Technopôle de Metz 2000 - Cescom - 4, rue Marconi - 57070 METZ (France)

Unité de Recherche INRIA Rennes IRISA, Campus Universitaire de Beaulieu 35042 RENNES Cedex (France)
Unité de Recherche INRIA Rhône-Alpes 46, avenue Félix Viallet - 38031 GRENOBLE Cedex (France)
Unité de Recherche INRIA Rocquencourt Domaine de Voluceau - Rocquencourt - B.P. 105 - 78153 LE CHESNAY Cedex (France)
Unité de Recherche INRIA Sophia Antipolis 2004, route des Lucioles - B.P. 93 - 06902 SOPHIA ANTIPOLIS Cedex (France)

EDITEUR
INRIA - Domaine de Voluceau - Rocquencourt - B.P. 105 - 78153 LE CHESNAY Cedex (France)

ISSN 0249 - 6399

RR.1989